DATA RATE ADJUSTER USING TRANSPORT LATENCY

[0001] This invention was made with Government support under Contract No. MDA 904-97-C-0613 awarded by the Maryland Procurement Office. The Government has certain rights in this invention.

BACKGROUND OF THE INVENTION

Field of the Invention:

[0002] The present invention relates to a method and apparatus for controlling the data rate in a network and more particularly to a method and apparatus for controlling the data rate between two processors on an Internet protocol network by adding a fixed latency

Discussion of the Background:

[0003] While a number of different types of network systems are in use, one of the common arrangements is a network based on the TCP/IP (transmission control protocol/internet protocol). This combined protocol is used to implement the Internet and also private networks such as Intranets. This arrangement provides for reliable delivery of sequenced packets of information across the Internet. An acknowledgment system is utilized so that the sending device will know that the data

has been received. A timing mechanism can be used so that if an acknowledgment is not received within a certain amount of time, the packet is retransmitted. This particular system is well known and widely used.

[0004] An example of such a system is shown in Figure 1 where an Internet protocol network 10 is shown connecting a client processor 12 with various server processors 14 and 16. The network includes routers 18 and 20 which control the flow of data on the network. One problem which can occur is the overloading of the router which causes it to crash when the data rate increases temporarily beyond the capability of the router. This problem has sometimes been addressed by attempting to control the data rate in the network.

[0005] U.S. Patent 6,115,357 which is assigned to Packeteer, Inc., describes the use of TCP/IP protocols and their use in networks. This device attempts to pace the data flow by arranging data transmission over a period of time after determining a round trip time for the data.

[0006] U.S. Patent 6,046,980, which is assigned to the same company, utilizes packet parsing based on content. That is, the packets are classified and a particular rule is applied according to their classification.

[0007] U.S. Patent 6,038,216, also assigned to Packeteer, Inc. shows data rate control without data rate supervision. Latency is added to the acknowledgment packet and the size of the flow control window is adjusted to control the data rate.

[0008] U.S. Patent 6,119,235 shows a method for quality of service management where the round trip time of the data is determined and delaying transmission of a receive acknowledgment signal when the bit rate is greater than a bit rate limit.

[0009] Similarly, other patents describe the measurement of the data flow in order to control in some fashion the data rate in a network.

[0010] While various solutions are possible for this type of problem, they often involve making major changes of some kind in either the client processor or the server processor which may be difficult and expensive. It would be advantageous to correct the problem merely by making adjustments of some sort in the network itself rather than requiring changes to either the client processor or server processors.

SUMMARY OF THE INVENTION

[0011] Accordingly, the present invention provides a method for adjusting the data rate in a network by adding a fixed delay thereto.

[0012] The present invention also provides an apparatus for delaying packet transport which reduces the data rate in a network by inserting a fixed delaying device.

[0013] The present invention also provides a system for adding fixed transport latency to selected data in a network in order to adjust the data rate downwardly in a network situation.

[0014] The present invention further provides a method for preventing a data rate overload by adding transport latency to the network which is determined by a low pass filtered data rate.

[0015] The present invention further provides an apparatus for adjusting the transport latency in a network by adding a delay device thereto whose delay is determined by a low pass filter of the data rate.

[0016] Briefly, this is accomplished by adding a delay processor to the network which causes a fixed delay of the data so that the data rate does not become too high, or to add a delay determined by a low pass filtered data rate.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017]   A more complete appreciation of the invention and many of the attendant advantages thereof will be readily obtained as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

[0018]   Figure 1 is a block diagram showing a prior art network arrangement;

[0019]   Figure 2 is a block diagram showing a network arrangement of the present invention;

[0020]   Figure 3 is a flow chart showing the algorithm used to control the delay processor;

[0021]   Figure 4 is a flow chart showing the algorithm used in a second embodiment.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0022]   Referring now to the drawings, wherein like reference numerals designate identical or corresponding parts throughout the several views, and more particularly to Figure 1 thereof, wherein a prior art network is shown. As described above, the network 10 has connected thereto a number of processors including a client processor 12 and server processors 14 and 16. The network has incorporated therein routers 18, 20 or other similar devices which control the flow of the data in the network. At such times that the data flow rate of the data stream increases temporarily, it is possible that the network will be overloaded or the router will be overloaded and thus will crash. The crash produces a number of deleterious effects including at least a temporary interruption in the data stream. Many of the times the increase in the data stream is driven by the TCP/IP algorithm searching for the maximum possible rate on

the network. In the past, attempts at solving these problems were aimed at the client or server processors. However, for the network providers, this is often a difficult or expensive proposition. Instead, it is desirable to provide the network with some means of handling this problem.

[0023]   Figure 2 is a block diagram which shows a network onto which a delay processor 30 has been added to solve this problem. In this system, data still flows from the client processor to the server processor and vice versa. However, the data is caused to flow through the delay processor when traveling in either direction. The delay processor does nothing but merely passes the data packets back to the network, but with a fixed delay. Using added delay to increase network latency limits the maximum TCP/IP algorithm rate to a value the network can tolerate.

[0024]   The particular amount of delay is either calculated in advance or determined in advance by experimentation. Once this has been set, it should not be necessary to adjust it, at least in the short run. As the latency or round trip delay increases, the data rate in the network will decrease. Thus, it is possible to control the data rate in the network independent of the client processor or server processors. This latency can be inserted in either direction or both directions if desired. By adding a fixed latency, an output stream is produced whose bandwidth will vary. It is also possible to specify by a packet selection list that certain packets will be delayed while others are not. In reality, the TCP/IP system will adjust the bandwidth based on the latency so that adding the fixed latency produces the required bandwidth adjustment. Thus, once the fixed delay is added the client TCP/IP system will handle the actual work of adjusting the data rate. While this will mean that if the latency does increase due to external network considerations, then the bandwidth will go lower than intended. However, it will still prevent a network traffic overload, which is the main purpose of this system.

[0025] Using a fixed delay avoids having to know the specific details of the TCP/IP protocol in use. This method will be successful with the newer TCP/IP protocol algorithm variations now coming into use. Implementation of a fixed delay is a simple operation thus allowing for increased processing capacity or reduced hardware requirements as appropriate over the more complex methods

[0026] The delay processor includes memory buffers into which the data is placed. The processor includes a clock circuit for generating timing signals and a controller for determining how long the data should remain in the buffers. The amount of delay is stored in a configuration table. The configuration table may be reset through the operators of the network. However, this will be done manually after observing the operation of the network, and it would not be adjusted on a short term basis by the instantaneous data rate. Data is placed into memory buffers and stored there for a set amount of time before being released. Thus, the round trip latency or the time it takes to make the delay, is increased by an amount which depends on the configuration table. Since TCP/IP is a reliable protocol which uses a limited capacity re-transmit buffer, the data rate will decrease as the latency increases. By utilizing this process, the latency can be increased in the data stream in either direction or both directions.

[0027] When utilizing this system, the problems of traffic overloads or crashing routers is reduced substantially. For example, when an office processor backup program achieves a data rate that overloads the office infrastructure router, in the past, the router crashed due to the increase in data rate. By adding latency using the delay processor, the data rate is reduced and crashing does not occur.

[0028] The delay processor may be a Sun Microsystems Unix Processor including a program which fixes the length of time the data should remain in a memory buffer before being returned to the data stream.

6

[0029]   Figure 3 is a flowchart which indicates the algorithm for controlling the movement of the data into and out of the memory buffers in the delay processor. Step 100 indicates the arrival of a signal, either a regularly occurring clock signal or the arrival of a data packet. Upon arrival, it is determined whether this is a clock signal or a data packet as indicated in step 102. If a new data packet has arrived, step 104 determines whether this is a configuration table packet or not. If it is, it is used to update the configuration table as indicated in step 106 to change the amount of delay and/or the packet selection list and the processor returns to step 100 to await the arrival of another signal. If it is not a configuration table packet, the processor determines whether the data packet is one to be delayed. The packet selection list indicates that certain packets are delayed and others not. This occurs in step 108. If the packet is not to be delayed, the packet is forwarded as indicated in step 112 and the processor returns to step 100 to await the next signal. If the packet is to be delayed, the packet is queued and placed in a memory buffer and the specific delay parameters are determined from the table in step 110.

[0030]   If the signal input is a regularly occurring clock signal as determined in step 102, the processor as indicated in step 114 reads whether there are any packets stored in the memory buffers. If all the packets have been processed so that the buffers are empty as indicated in step 116, the processor awaits the arrival of the next signal at step 100. If some packets remain in the buffers, the processor proceeds to step 118 which determines whether the first stored packet has reached its time limit. If not, the processor returns to step 114, to look at the next stored packet. If it is time to send the packet, the processor in step 120 removes the packet from the buffer and forwards the packet in step 122. The processor then returns to step 114 to look at the next stored packet.

[0031]   When the packet is originally stored in step 110, the configuration

table is consulted to determine the specified amount of time that the packet should remain in the memory buffers. The time at which it may be released is stored along with the data so that when clock signals are received, each packet may be reviewed to determine if it should be released.

[0032]   While the invention described above utilizes a fixed delay which is only changed in the configuration table by manual operation of the operator, it also would be possible to utilize this system by measuring the data rate in a fashion somewhat similar to that of the prior art, but instead of utilizing the instantaneous rate itself to control the flow, rather the data rate is low pass filtered to provide a very slow change to the configuration table so that the delay may be slowly adjusted over a long period of time. This would avoid the necessity of the operator to manually change the configuration table but at the same time still gains the advantages of applying a fixed delay. The operation would be different from the prior art which uses an instantaneous measurement and change and in fact the data packets would operate in the same fashion as a fixed configuration table arrangement because of the slow change relative to the packet time. Using a slowly varying delay value avoids being misled by rapidly changing network latency conditions due to network overload.

[0033]   Figure 4 is a flow chart of the algorithm which should be followed for this modified arrangement. Only the right side of Figure 3 is changed and accordingly the left side is not shown in Figure 4. When an IP packet is received and typed in step 102 it is passed to determine if it is a configuration packet or not. If it is a manual change, it is updated in step 106 just as in Figure 3. If it is not, the system proceeds to collect data rate statistics. If the IP packet is one of those which is listed for being delayed as determined in step 108, it is queued in step 110 just as in Figure 3. However, new step 126 then computes the data rate based on the static's collected in step 124. The data rate is low pass filtered in step 128 and the delay which is queued

along with the packet is then adjusted if it is necessary. The system then returns to step 100 just as in Figure 3. If the IP packet is not on the list for delay, it is forwarded in step 112 just as in Figure 3.

[0034]   Thus, the difference between the arrangement of Figure 3 and that of Figure 4 is that the second embodiment collects the rate statistics computes the data rate and low pass filters it. The delay is then adjusted by this amount when the packet is placed in a queue.

[0035]   Other configurations of the data rate computation could also be envisioned. For example if it is desired to not rely on manual operation at all, steps 104 and 106 can be eliminated and the data rate computation could be placed there with a default value set in the system. Also, it would be possible to move steps 126, 128 and 130 to branch off of step 124 so that the data rate is computed filtered and adjusted in parallel to the branch including step 106. In this arrangement the data rate in the configuration table is adjusted separately from the queuing of the individual packet.

[0036]   Thus, it is seen that it is possible to remove a potential traffic overload or crash problem in the network by deliberately adding a delay to the data stream which is otherwise not necessary. In doing so, the data rate is reduced to the point that the routers and other hardware in the network are not overwhelmed. This idea is contrary to the usual view of increasing data rates as much as possible. Thus, this acts as a type of data rate averaging device which avoids temporary increases to provide better data flow.

[0037]   Numerous additional modifications and variations of the present invention are possible in light of the above teachings. It is therefore to be understood that within the scope of the appended claims, the invention may be practiced otherwise than as specifically described herein.